2024 겨울 세미나

Zero-/Few-Shot Anomaly Detection

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

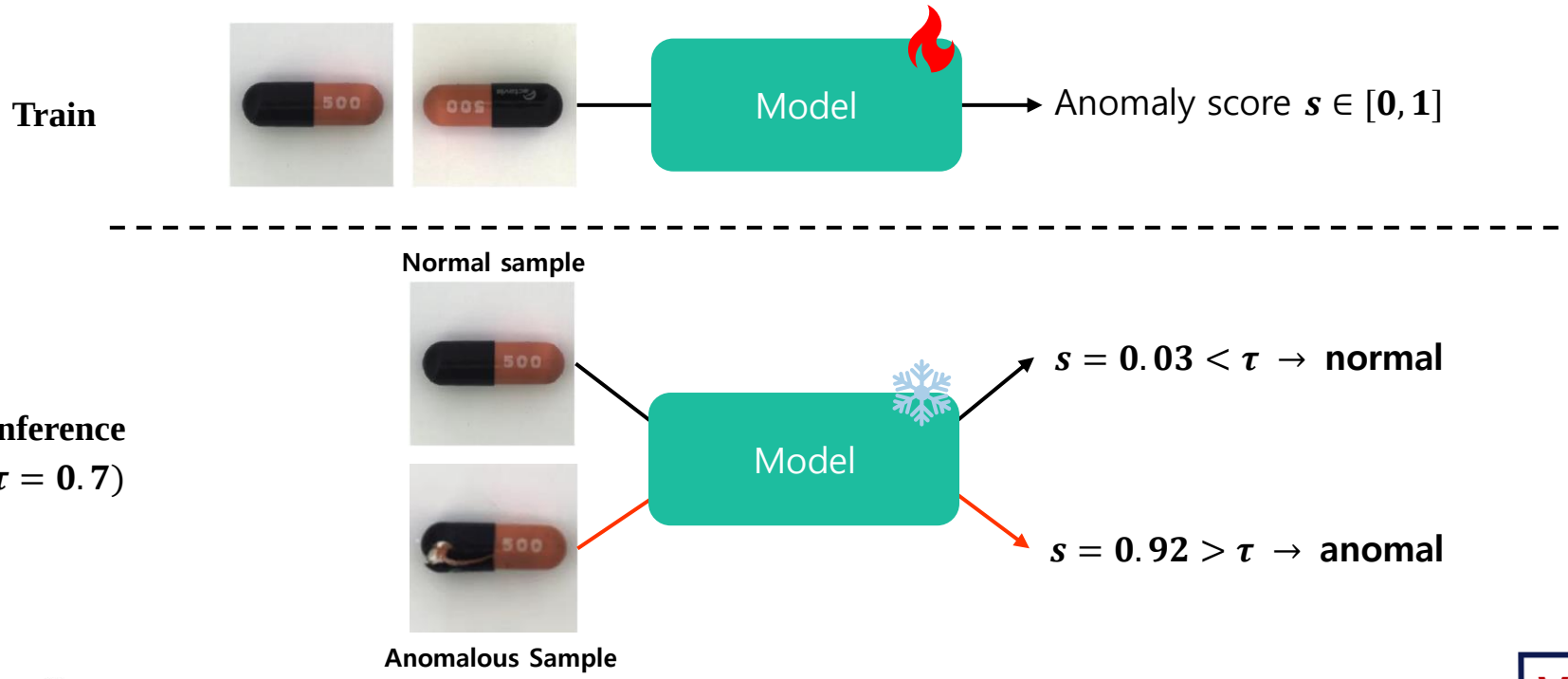
Presented By
김현빈

Contents

- Introduction
  - Anomaly Detection
  - Zero-shot / Few-shot Learning
    - Contrastive Learning
    - CLIP
    - LLM
- Paper Review
  - WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation (CVPR 2023)

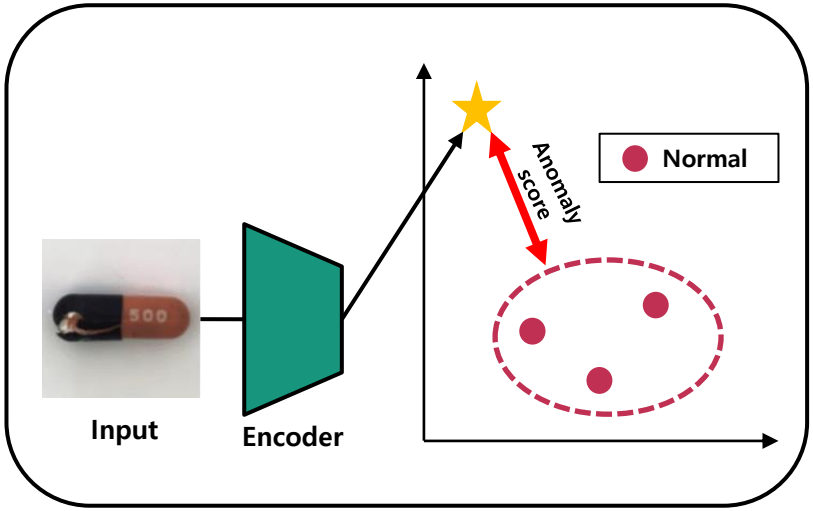
Introduction

- Anomaly Detection
  - Anomaly Detection은 Input sample의 anomaly score를 계산하고, 이를 바탕으로 normal, anomaly를 구분하는 작업
  - 모델은 training 과정에서 normal sample의 feature을 학습하고, Inference 과정에서 새로운 입력과 학습된 normal feature의 차이를 바탕으로 anomaly score를 계산

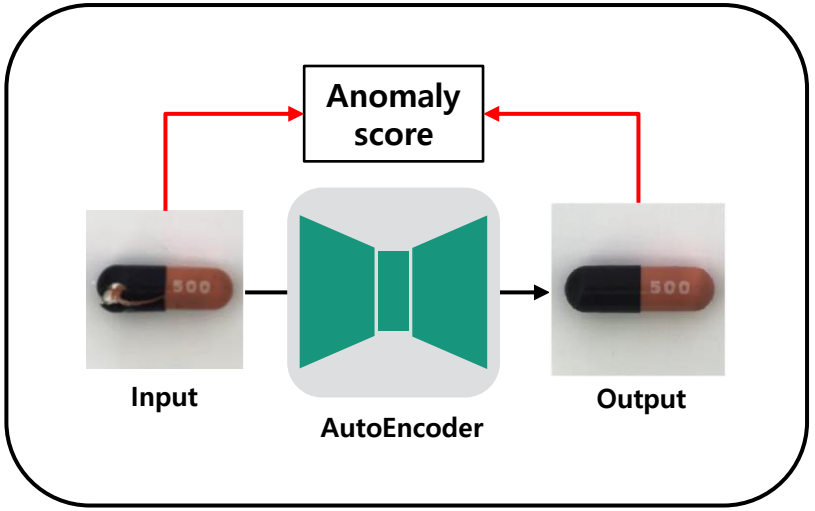


Introduction

- Anomaly Scoring Model
  - Anomaly score를 계산하는 다양한 방법론이 제시됨
  - Embedding based, reconstruction based method가 일반적으로 많이 사용됨
    - Embedding based: Embedding space에서 학습에 활용했던 normal sample과의 거리를 이용
    - Reconstruction based: Normal sample에 대해서만 잘 복원하도록 학습된 모델을 활용. Input과 output의 차이를 바탕으로 anomaly score로 계산



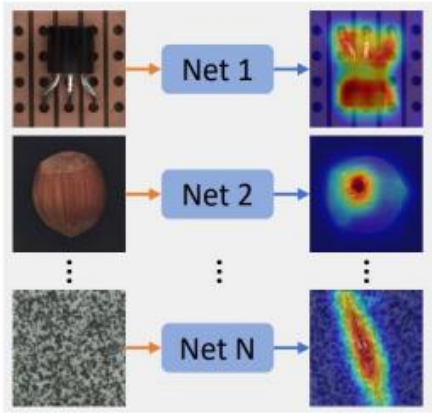
Embedding based
(PatchSVDD, PatchCore)



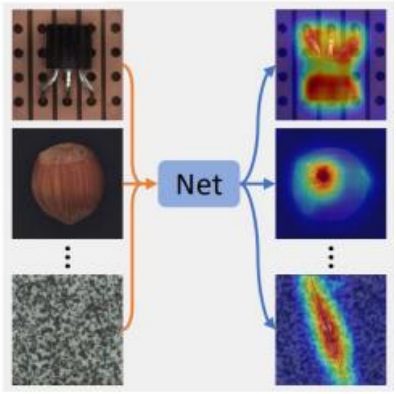
Reconstruction based
(InTra, AnoDDPM)

Introduction

- Anomaly Detection
  - Few- / Zero-normal shot Setting
    - 기존에 제시된 방법들은 한 개의 object/class에 대하여 개별적인 모델을 학습하는 one-class 방식으로 작동함
    - 하지만 class 개수가 증가하면 큰 memory가 소모되는 등의 문제점이 존재, 이를 개선하고자 여러 class의 대응할 수 있는 unified model이 등장함
    - 이를 넘어서서, 처음보는 class의 normal sample을 조금만 학습하거나 아예 학습하지 않아도 해당 class에 대응할 수 있는 Model이 제시됨



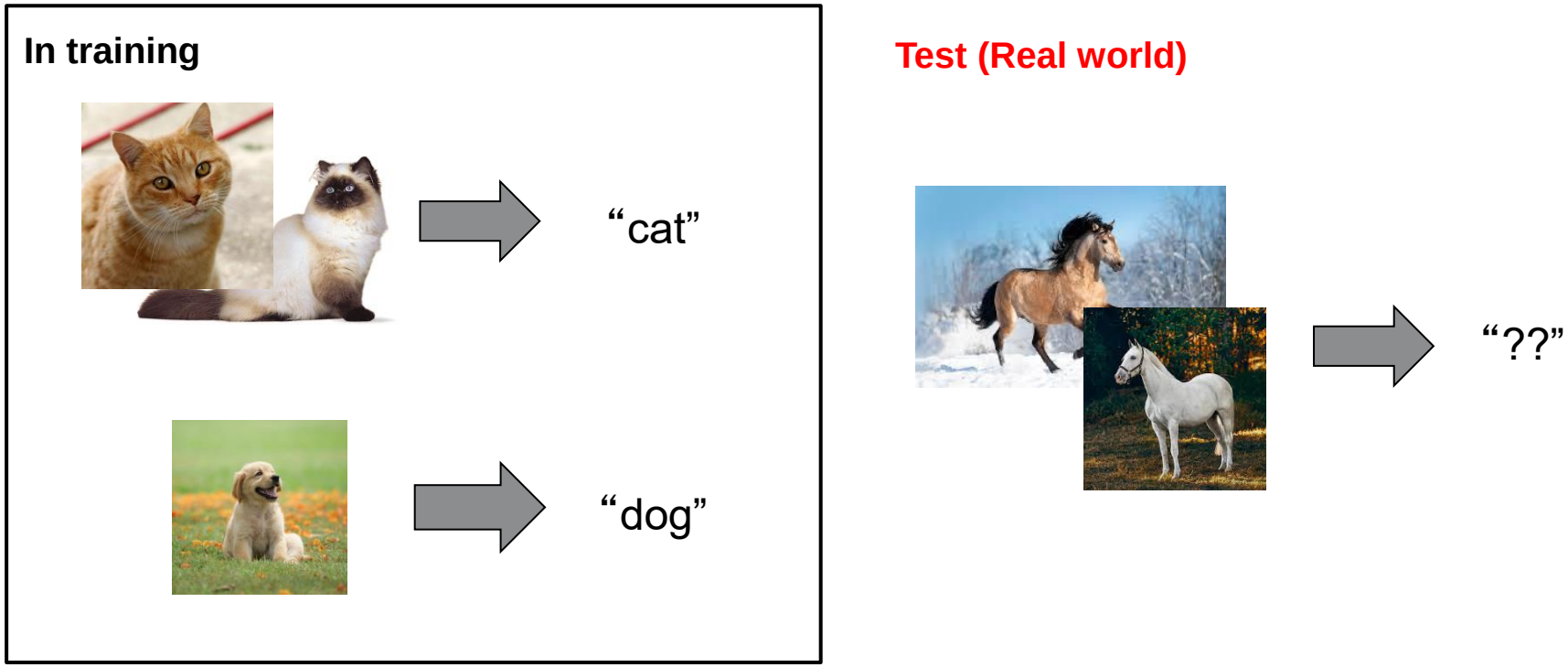
<One-class-one-model scheme>



<Unified model scheme>

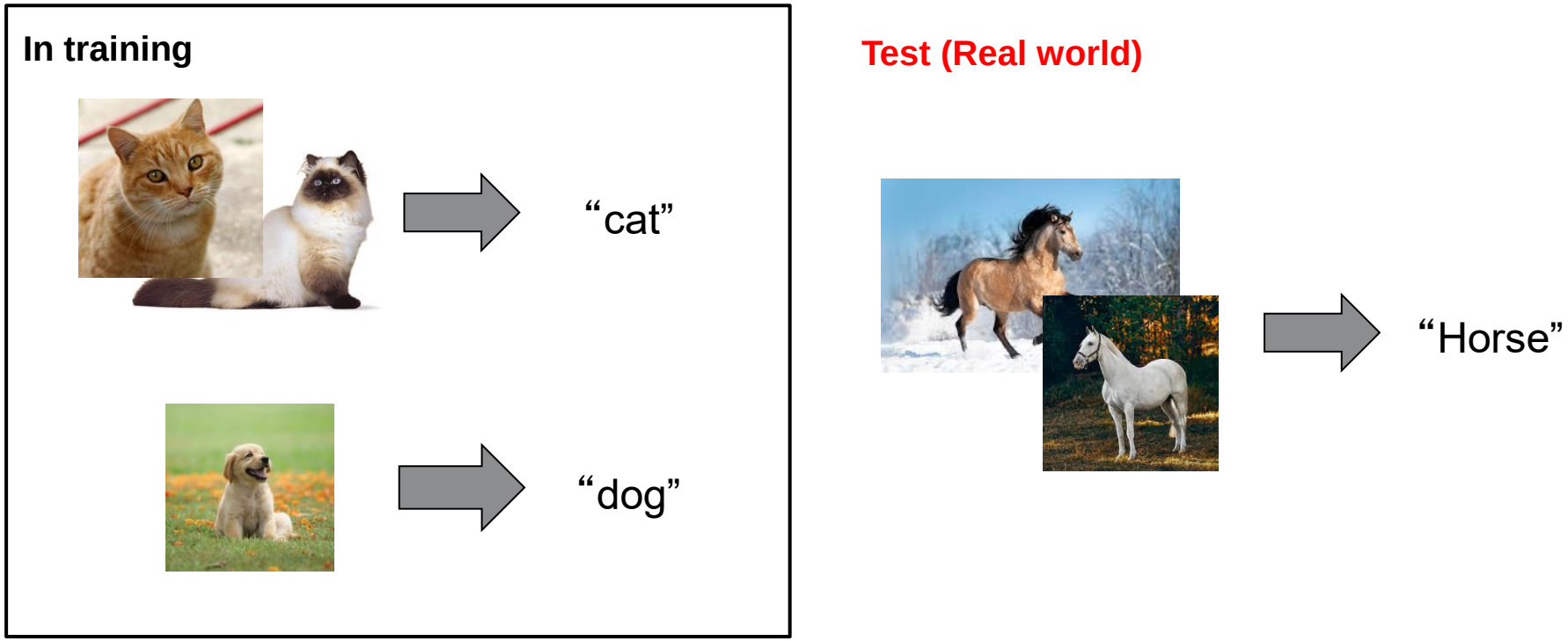
Introduction

- Few-shot / Zero-shot Learning
  - Few-shot / Zero-shot Learning in classification task
    - 기존 모델은 학습된 클래스만 구분할 수 있지만, real world에서는 학습하지 않은 클래스가 주어지면 모델이 잘 분류하지 못한다는 문제가 있음



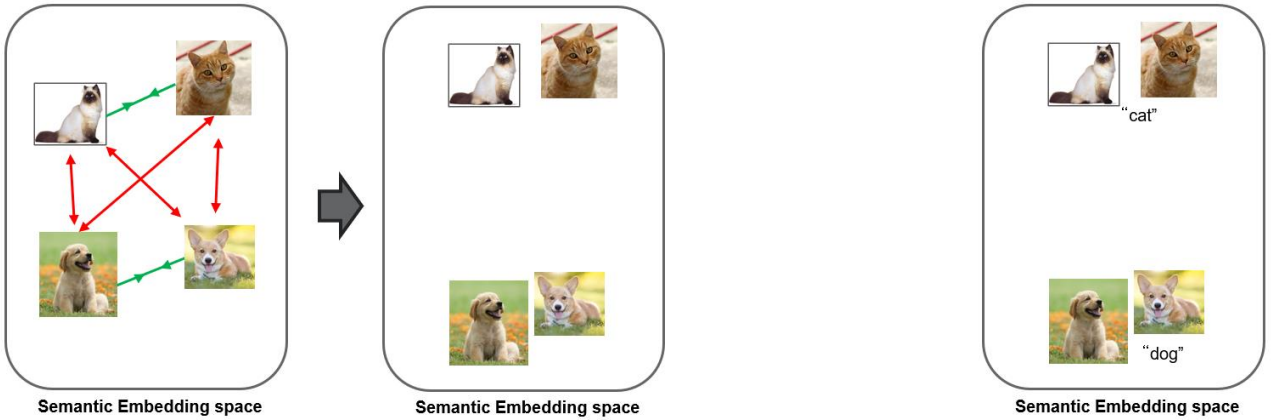
Introduction

- Few-shot / Zero-shot Learning
  - Few-shot / Zero-shot Learning in classification task
    - 처음보는 클래스의 sample을 몇 장만 학습하거나 / 아예 학습하지 않아도 해당 클래스에 대응할 수 있도록 모델을 학습시키는 방법을 Few-shot / Zero-shot Learning이라고 함



Introduction

- Few-shot / Zero-shot Learning
  - Contrastive learning
    - 모델이 같은 클래스는 가깝게, 다른 클래스는 멀어지도록 매핑하게 학습하여 모델이 일반적인 특징을 추출하게 하는 방법
    - 일반적인 특징을 추출할 수 있어 새로운 클래스에 대응할 수 있음
  - Multi-modal contrastive learning(Visual&textual)
    - Visual feature와 text feature를 같은 embedding space에 mapping하여 contrastive learning을 수행하는 방법.
    - 이후 소개할 CLIP에서 활용하며, 좀 더 의미 있는 feature를 추출할 수 있음

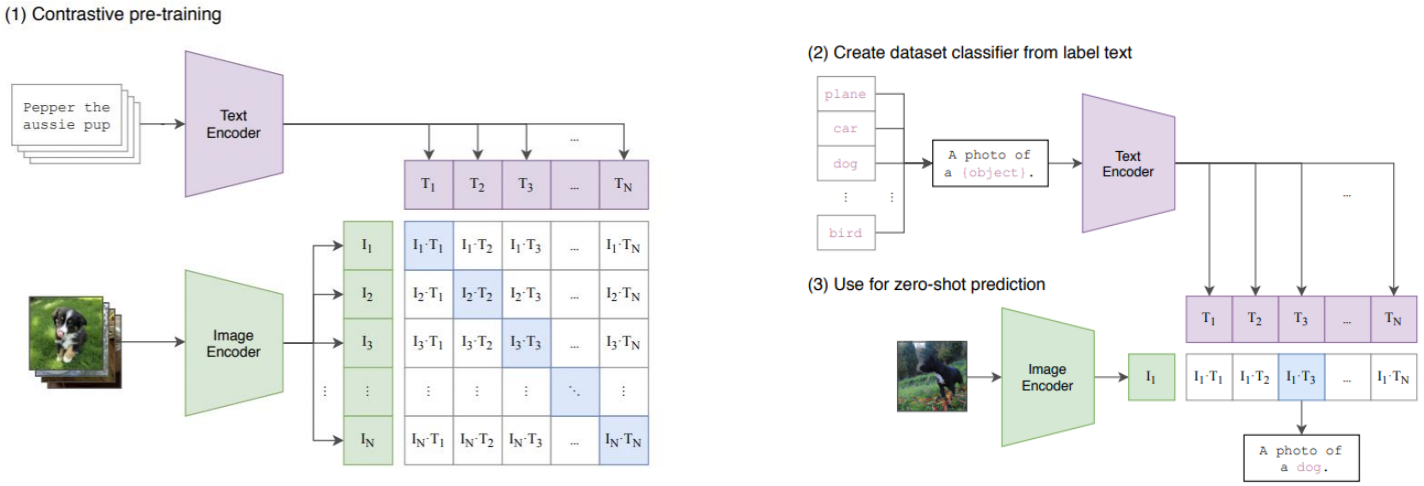


Contrastive learning

Multi-modal Contrastive learning

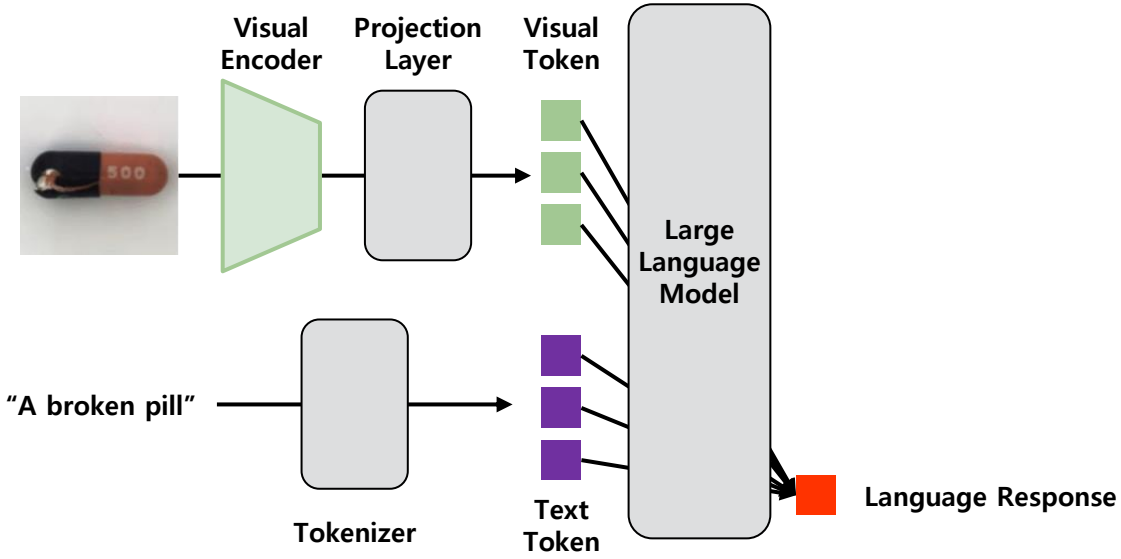
Introduction

- Few-shot / Zero-shot Learning
  - CLIP
    - Image embedding과 text embedding 간의 contrastive learning을 통해 모델을 학습
    - 자연어를 guidance로 활용하여, image encoder가 좀 더 의미 있는 feature를 추출하도록 학습
    - Large-scale dataset으로 학습하여 강력한 representation 능력을 갖고 있어, zero-shot vision task에서 활용 가능



Introduction

- Few-shot / Zero-shot Learning
  - Utilizing LLM (Large Language Models) for downstream tasks
    - CLIP의 경우 각 encoder에서 획득한 Image embedding과 text embedding이 가까워지게 학습할 뿐, 두 modality의 정보가 공유되지 못한다는 한계가 있음
    - Pretrained-Vision encoder의 last layer 대신 learnable projection layer를 추가하여 LLM에 호환될 수 있도록 변경
    - LLM이 각 modality를 동시에 반영하며 학습하기에, 더 강력한 representation 능력을 가질 수 있음

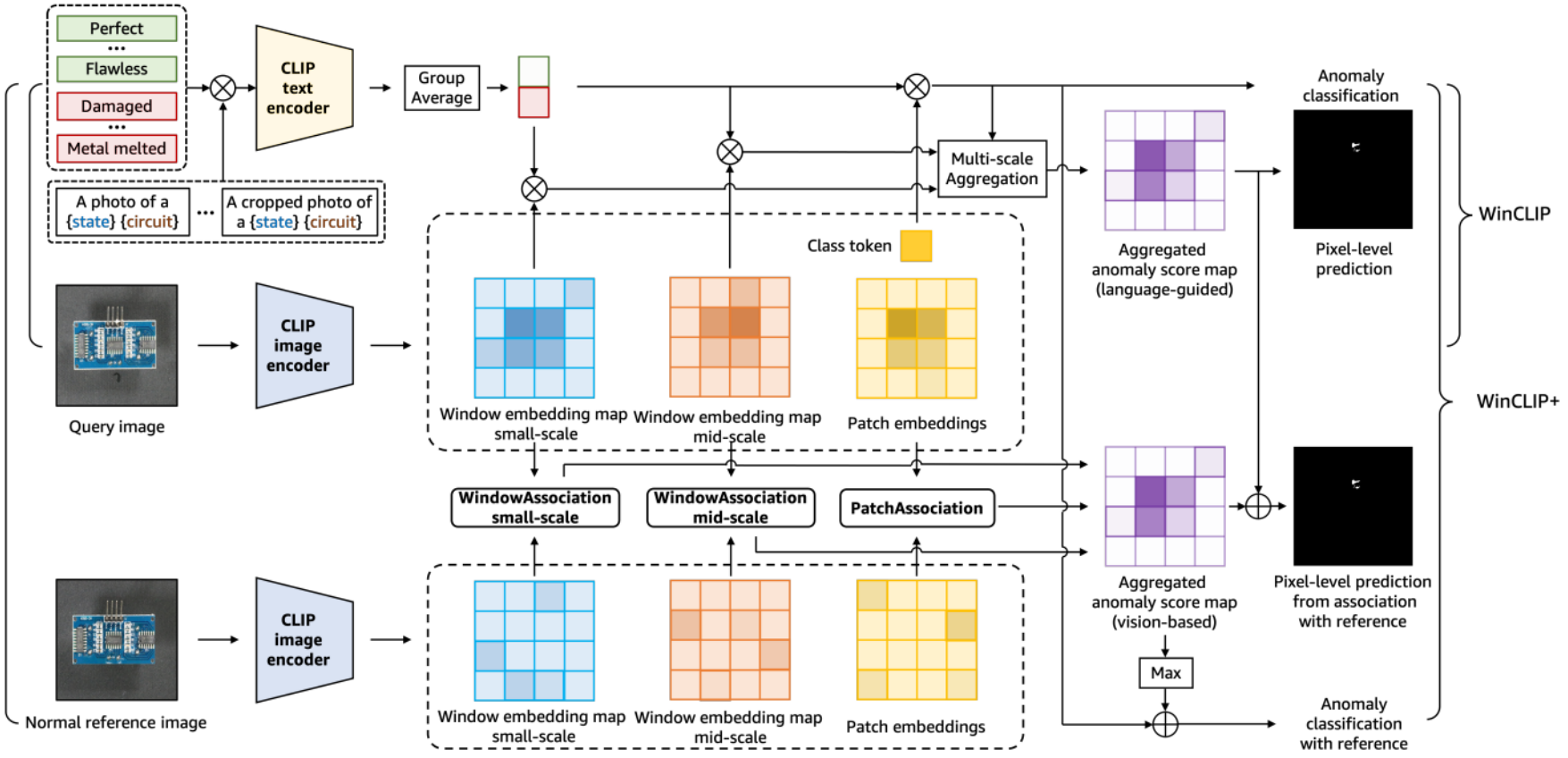


Paper Review

WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation (CVPR 2023)

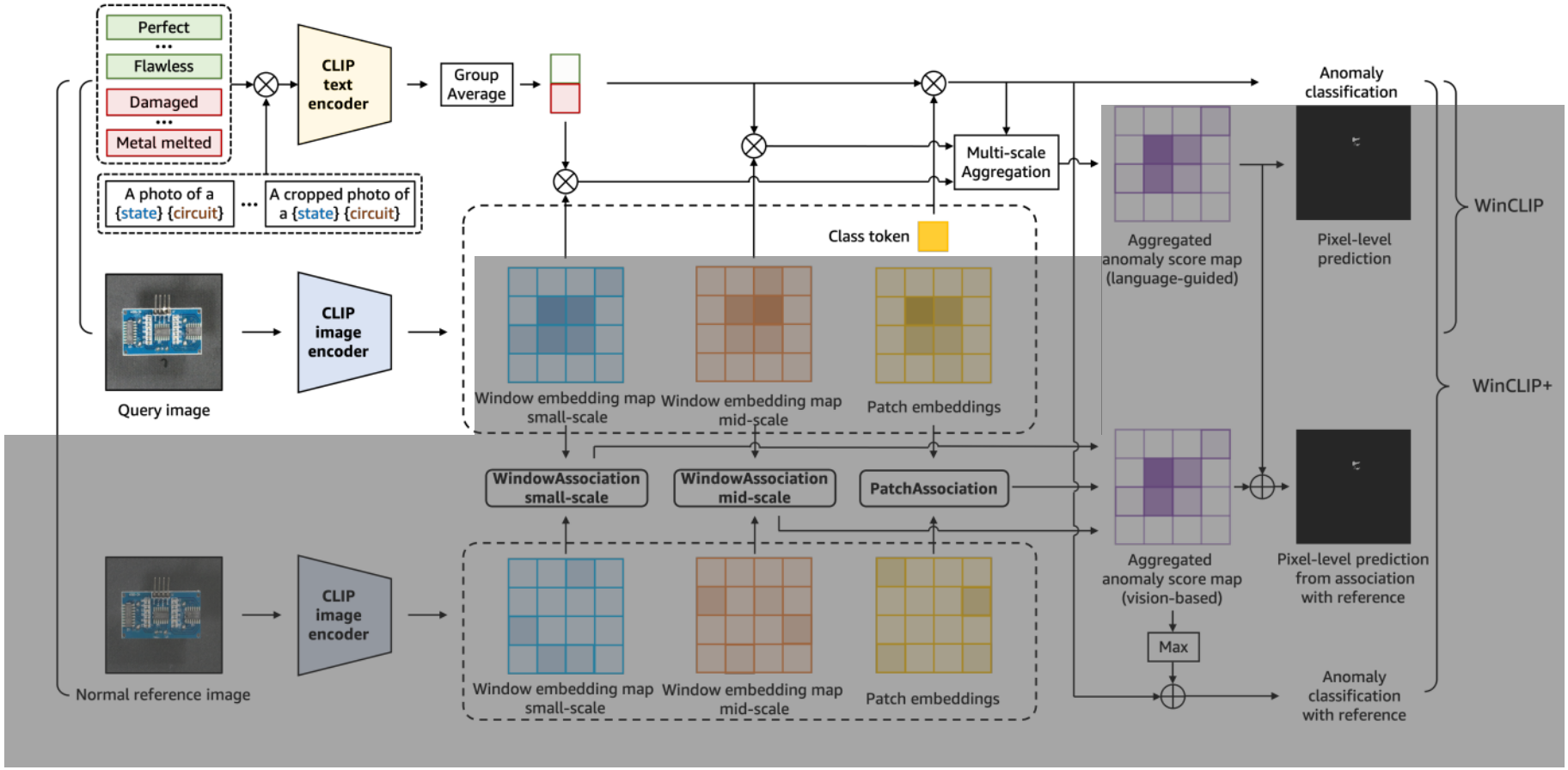
Method

- Overview



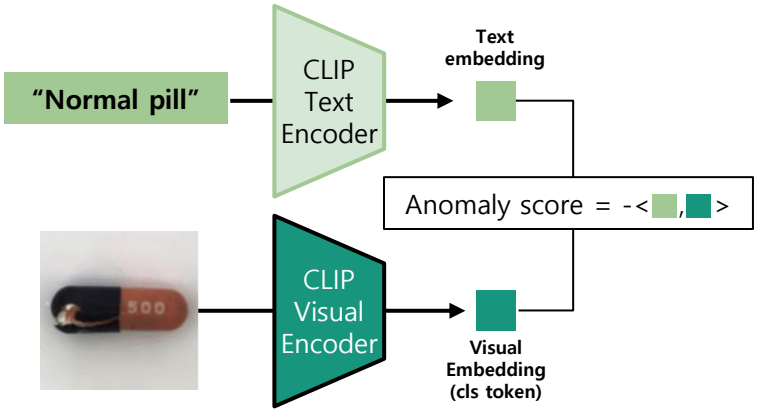
Method

- Language-driven Zero-shot Anomaly Classification

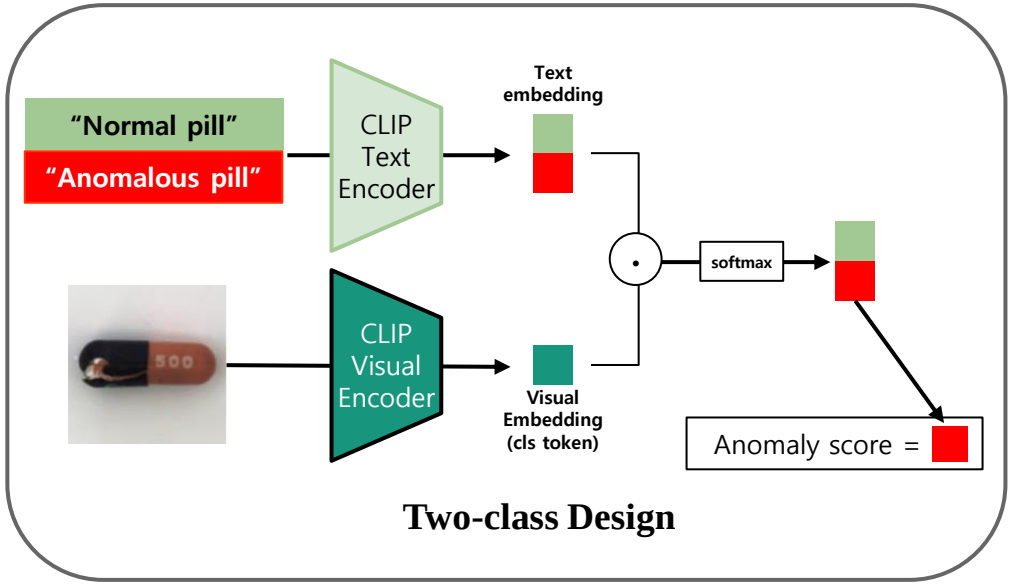


Method

- Language-driven Zero-shot Anomaly Classification
  - Two-class design
    - CLIP은 text와 visual feature를 같은 space에 embedding하도록 학습했기 때문에, 이를 이용해 Zero-shot Anomaly Detection을 수행할 수 있음
    - Pretrained CLIP model을 이용하여 image와 **Normal을 의미하는 text**와 **Anomaly를 의미하는 text**와의 similarity를 각각 계산하고, softmax를 취한 값을 anomaly score로 활용



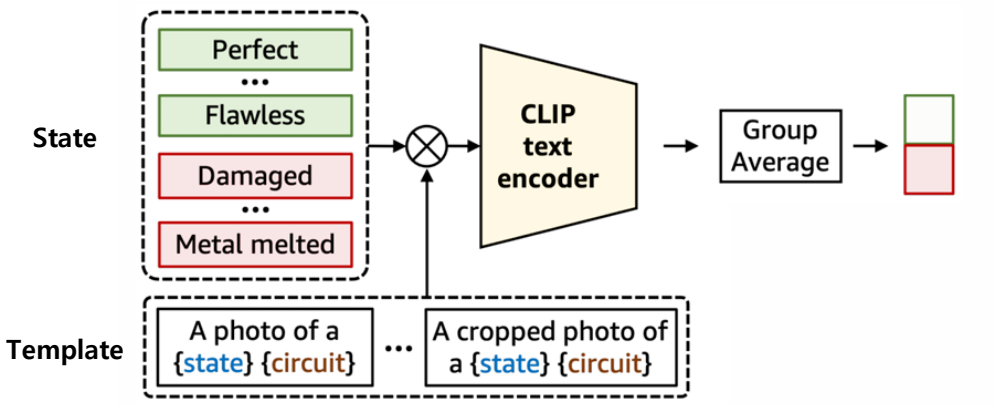
One-class Design



Two-class Design

Method

- Language-driven Zero-shot Anomaly Classification
  - Compositional prompt ensemble (CPE)
    - Anomaly는 class에 따라 다양하게 발생하여 일반적인 text prompt로 처리하는데 한계가 있음
    - 이에 대응하기 위해, object의 상태에 대한 정의를 명확히 하는 방법을 제시
    - Object class별로 다양한 **state**와 **template**을 조합하여 다양한 text prompt를 생성하고, text encoder로부터 얻은 embedding들을 그룹별 (Normal/Anomalous)로 average하여 각 그룹을 대표하는 text embedding 생성
    - Two-class design으로 anomaly score를 계산하고, 이를 이용해 Anomaly Classification 수행



CPE scheme

Method

- Language-driven Zero-shot Anomaly Classification
  - Compositional prompt ensemble (CPE)

(a) *State*-level (normal)

- c := "[o]"
- c := "flawless [o]"
- c := "perfect [o]"
- c := "unblemished [o]"
- c := "[o] without flaw"
- c := "[o] without defect"
- c := "[o] without damage"

(b) *State*-level (anomaly)

- c := "damaged [o]"
- c := "[o] with flaw"
- c := "[o] with defect"
- c := "[o] with damage"

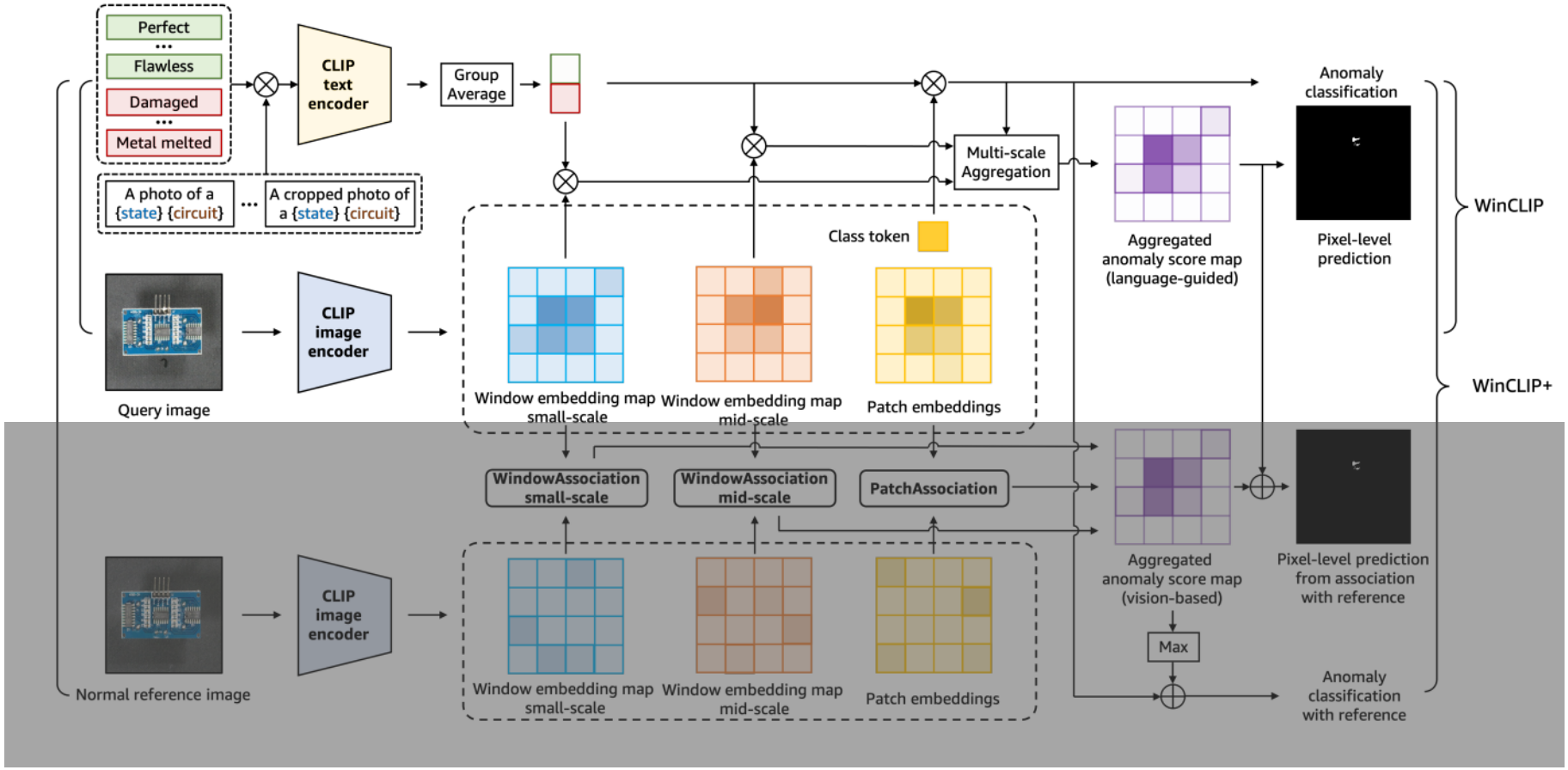
(c) *Template*-level

- "a cropped photo of the [c]."
- "a cropped photo of a [c]."
- "a close-up photo of a [c]."
- "a close-up photo of the [c]."
- "a bright photo of a [c]."
- "a bright photo of the [c]."
- "a dark photo of the [c]."
- "a dark photo of a [c]."
- "a jpeg corrupted photo of a [c]."
- "a jpeg corrupted photo of the [c]."
- (cont'd) "a blurry photo of the [c]."
- "a blurry photo of a [c]."
- "a photo of a [c]."
- "a photo of the [c]."
- "a photo of a small [c]."
- "a photo of the small [c]."
- "a photo of a large [c]."
- "a photo of the large [c]."
- "a photo of the [c] for visual inspection."
- "a photo of a [c] for visual inspection."
- "a photo of the [c] for anomaly detection."
- "a photo of a [c] for anomaly detection."

CPE example

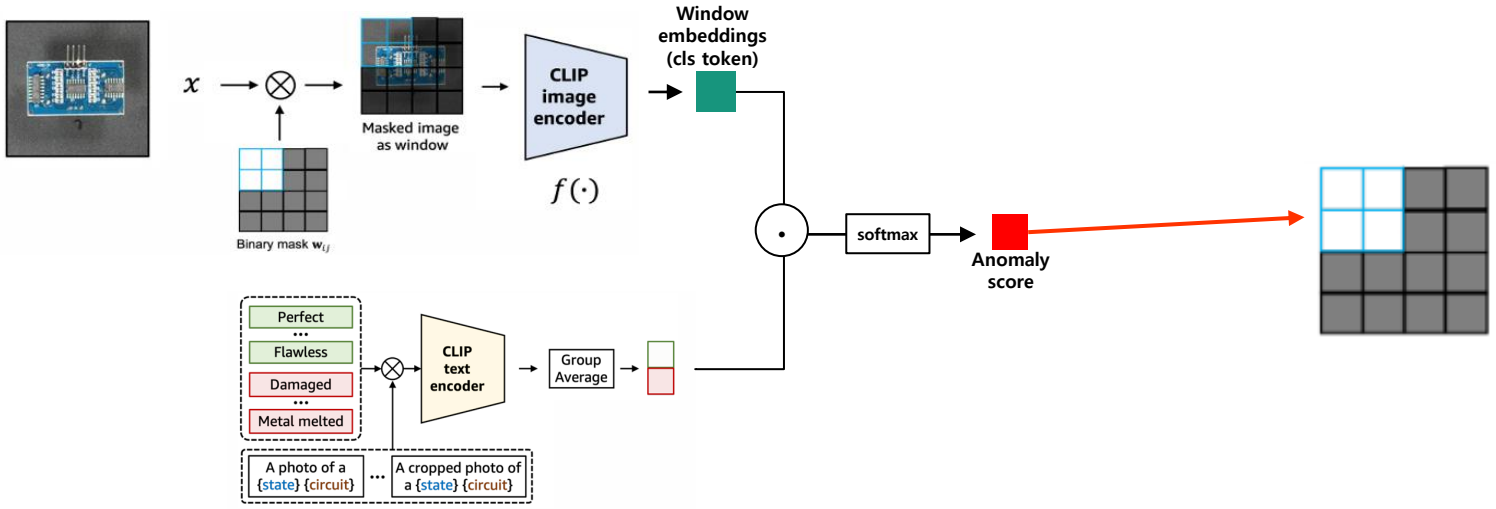
Method

- WinCLIP for zero-shot Anomaly Segmentation



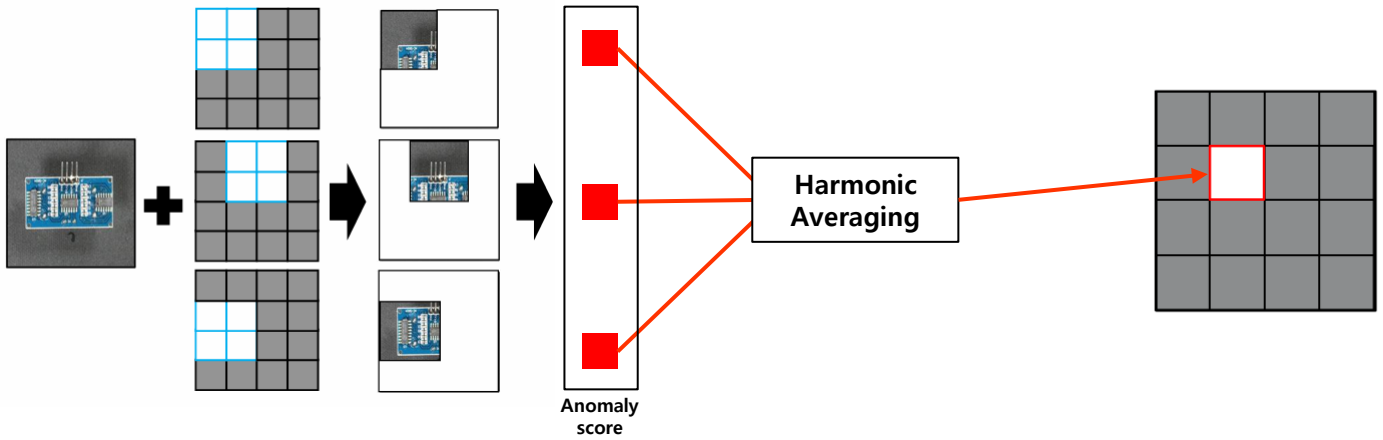
Method

- WinCLIP for zero-shot Anomaly Segmentation
  - Getting dense representation
    - Anomaly Segmentation을 수행하기 위해서는 local detail feature가 필요함
    - Patch-level sliding window를 활용하여, window를 제외한 나머지 부분이 **masking**된 이미지의 window embedding (**CLS token**)을 추출
    - 각 window로부터 얻은 embedding과, CPE로 얻은 text embedding의 similarity를 계산하여 anomaly score를 획득
    - 각 Window 내에 존재하는 모든 픽셀에 해당 window로부터 계산한 anomaly score를 부여



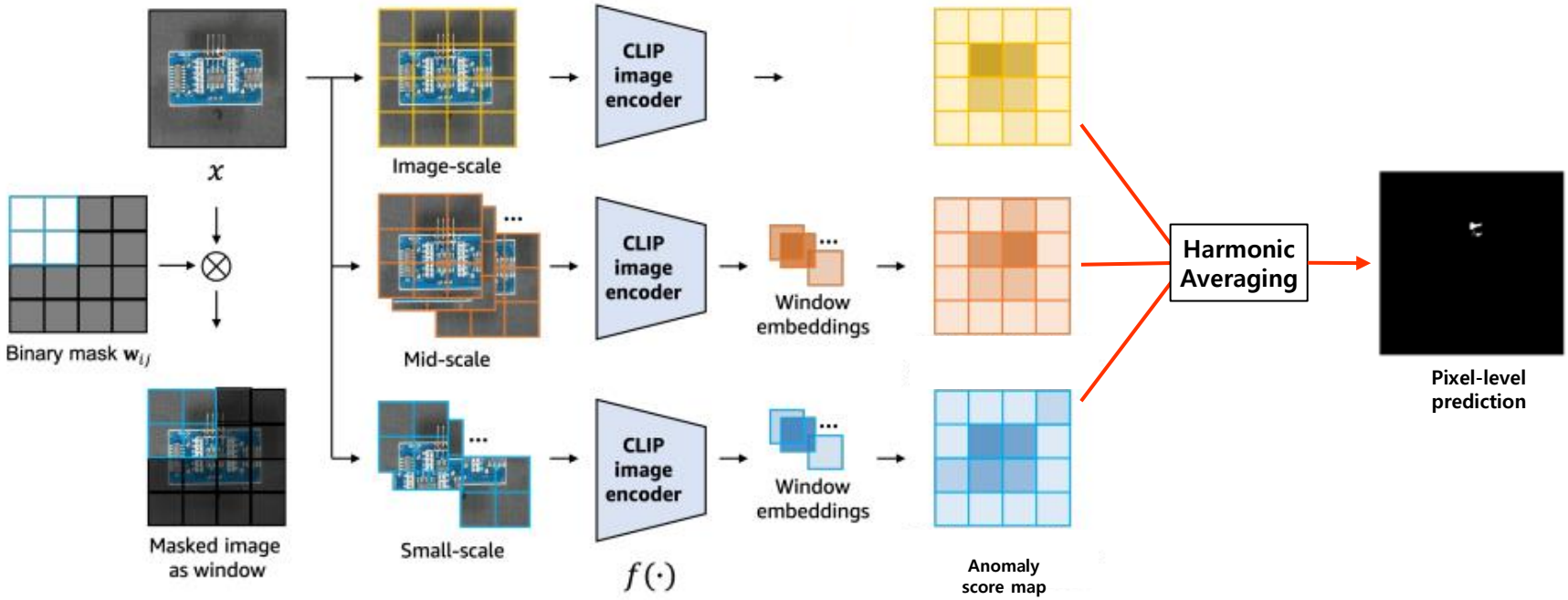
Method

- WinCLIP for zero-shot Anomaly Segmentation
  - Getting dense representation
    - Window 내에 존재하는 모든 픽셀에 anomaly score를 부여하면서, window가 겹치는 부분에 여러 번 score가 할당이 됨
    - Harmonic averaging을 취해 이들을 하나의 값으로 만들어 줌



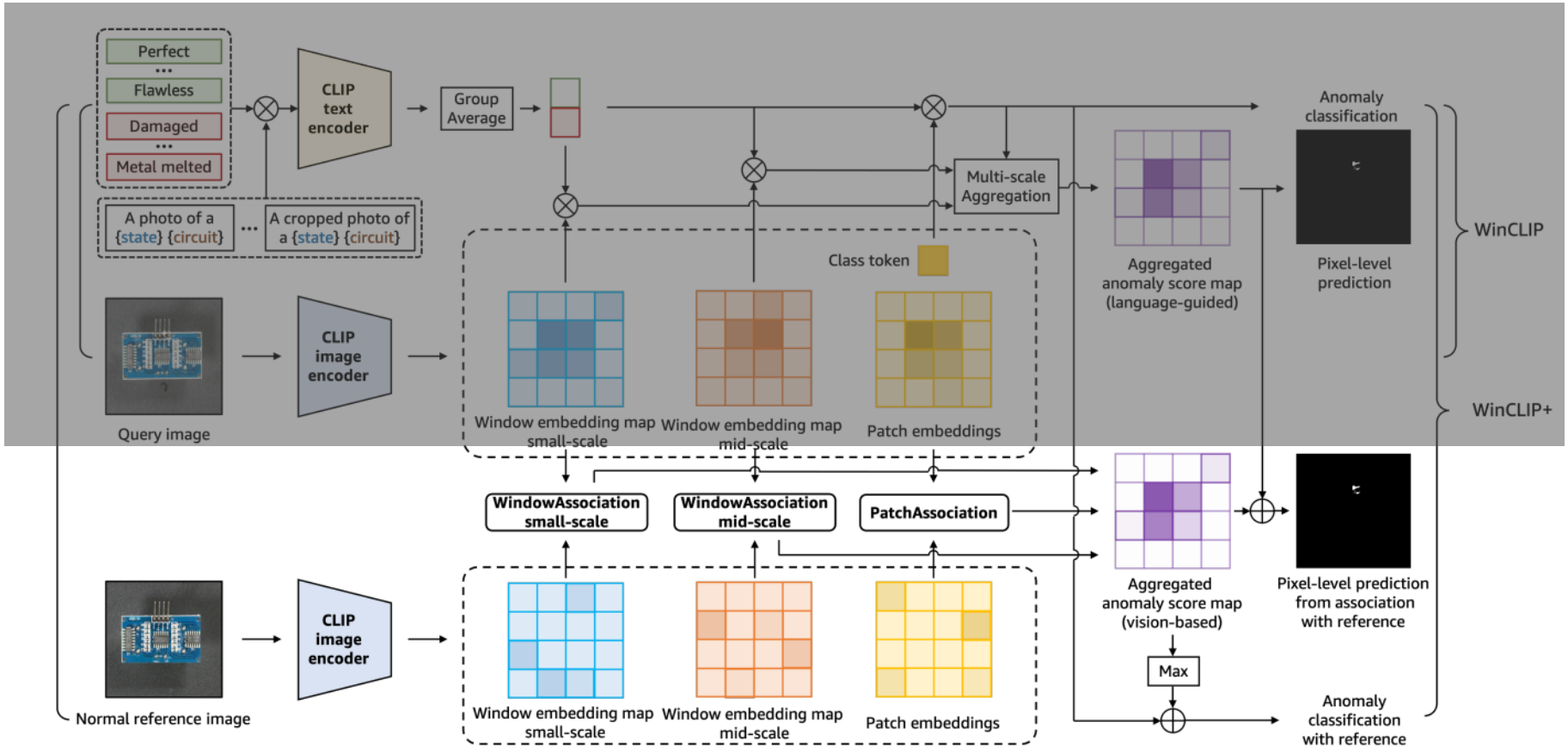
Method

- WinCLIP for zero-shot Anomaly Segmentation
  - Getting dense representation
    - WinCLIP은 small-scale(2x2) / mid-scale(3x3) / image scale로 나누어 앞선 과정을 수행함
    - Image scale의 경우 classification token이 아닌 penultimate feature를 사용
    - 각 Scale에서 얻은 score map에 harmonic averaging을 취해 최종 anomaly score map 획득



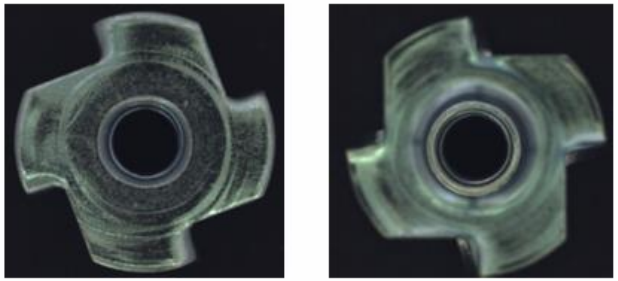
Method

- WinCLIP+ for few-shot Anomaly Segmentation



Method

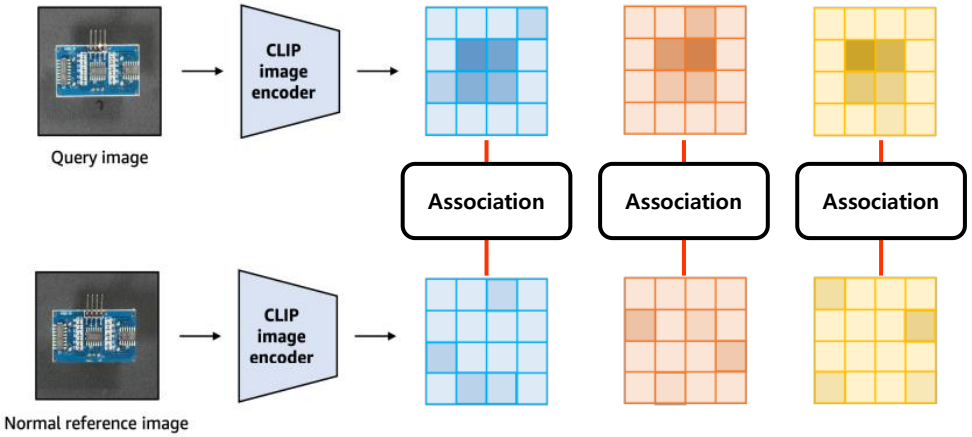
- WinCLIP+ for few-shot Anomaly Detection
  - Language guidance만으로는 anomaly classification/segmentation하기에 충분하지 않은 경우가 있음
  - 특정 defection은 visual reference를 통해서만 정의가 가능
    - Metal_nut: flipped upside-down, 정방향이 어딘지 reference가 존재해야 정의가 가능
  - WinCLIP+은 Few-normal shot을 활용하여, 좀 더 정확하게 anomaly를 구분할 수 있는 방법을 제시



Normal Anomaly

Method

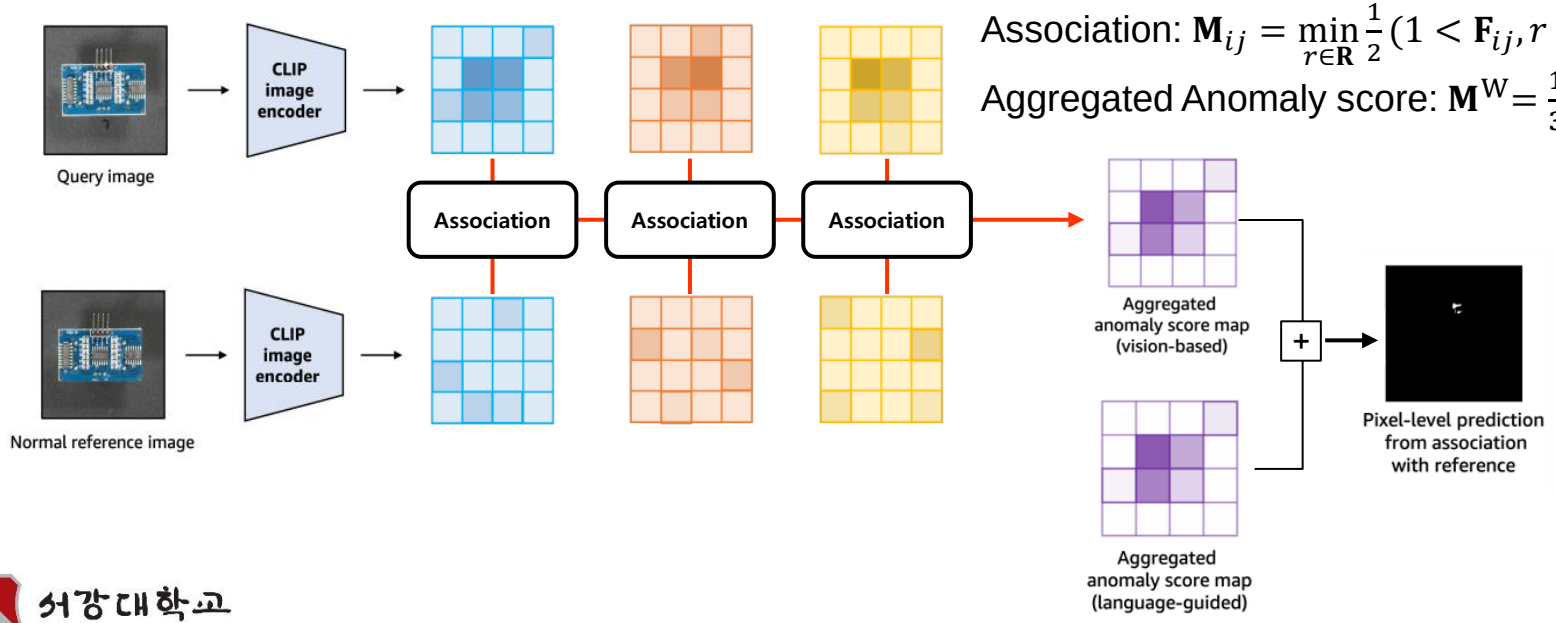
- WinCLIP+ for few-shot Anomaly Detection
  - Few-shot Anomaly Segmentation
    - Normal reference image들의 scale 별 feature를 저장
    - Query image와 저장된 모든 reference image들의 feature를 patch-wise로 비교
    - 가장 큰 유사도를 갖는 경우의 anomaly score를 해당 patch의 anomaly score로 사용



Association: $\mathbf{M}_{ij} = \min_{r \in \mathbf{R}} \frac{1}{2}(1 < \mathbf{F}_{ij}, r >)$

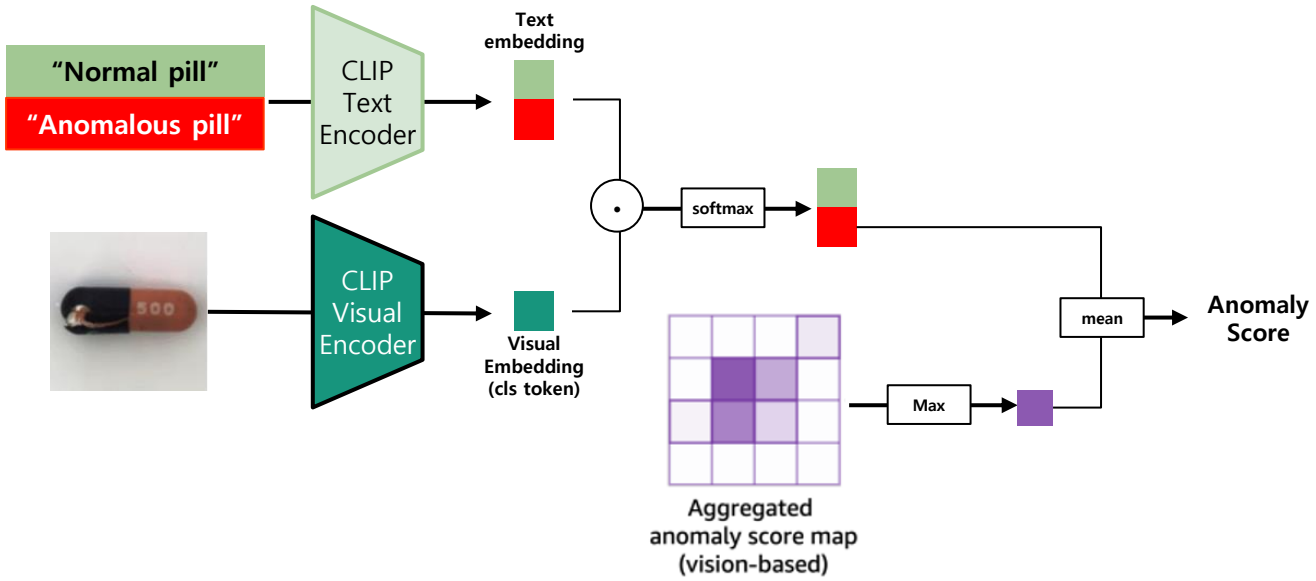
Method

- WinCLIP+ for few-shot Anomaly Detection
  - Few-shot Anomaly Segmentation
    - Scale별로 anomaly score를 계산한 후, 이를 average하여 visual anomaly score map을 생성
    - Text prompts와 비교하여 얻은 anomaly score map과 fusing하여 최종 Anomaly map 생성



Method

- WinCLIP+ for few-shot Anomaly Detection
  - Few-shot Anomaly Classification
    - Visual anomaly score map의 maximum value와 zero-shot classification과 같은 방법으로 계산한 language-guided anomaly score를 함께 사용



Experiment

- Zero-/Few-shot Anomaly Classification
  - Zero-shot Setting

Anomaly Classification		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F_1-max	AUROC	AUPR	F_1-max
0-shot	CLIP-AC [27]	74.0±0.0	89.1±0.0	88.5±0.0	59.3±0.0	67.0±0.0	74.4±0.0
	+ Prompt ens. [27]	74.1±0.0	89.5±0.0	87.8±0.0	58.2±0.0	66.4±0.0	74.0±0.0
	WinCLIP (ours)	**91.8±0.0**	**96.5±0.0**	**92.9±0.0**	**78.1±0.0**	**81.2±0.0**	**79.0±0.0**

- CLIP-AC : {“anomalous [obj]”, “normal [obj]”}만을 prompt로 사용
- Prompt ens. : Text template만을 변화하여 text embedding을 추출 후, 이들의 평균을 사용
- WinCLIP : State, Text template를 모두 변화시키는 CPE를 사용

- 실험 결과, CPE를 사용했을 때 더 좋은 성능을 내고 있음을 볼 수 있음

Experiment

- Zero-/Few-shot Anomaly Classification
  - Few-shot setting
    - Zero-shot setting에서 성능이 SOTA의 few-shot setting에서 성능보다 나은 것을 볼 수 있음
    - Few-shot setting에서도 SOTA보다 좋은 성능을 내는 것을 볼 수 있음

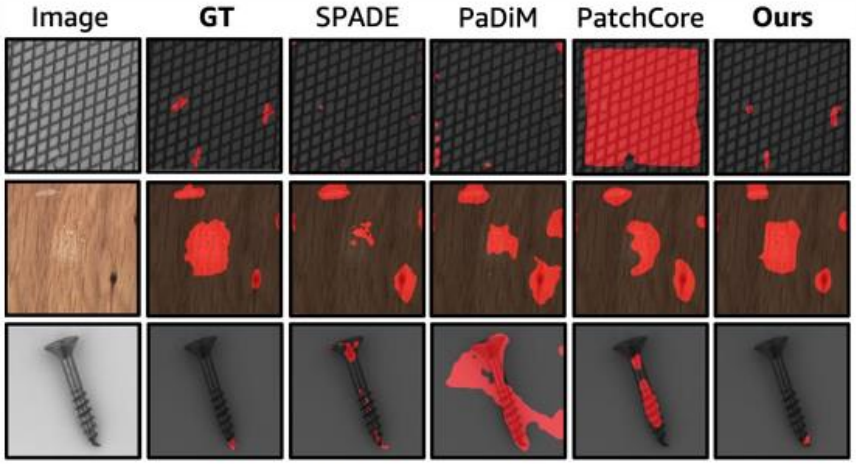
Anomaly Classification		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F_1-max	AUROC	AUPR	F_1-max
0-shot	CLIP-AC [27]	74.0±0.0	89.1±0.0	88.5±0.0	59.3±0.0	67.0±0.0	74.4±0.0
	+ Prompt ens. [27]	74.1±0.0	89.5±0.0	87.8±0.0	58.2±0.0	66.4±0.0	74.0±0.0
	WinCLIP (ours)	**91.8±0.0**	**96.5±0.0**	**92.9±0.0**	**78.1±0.0**	**81.2±0.0**	**79.0±0.0**
1-shot	SPADE [7]	81.0±2.0	90.6±0.8	90.3±0.8	79.5±4.0	82.0±3.3	80.7±1.9
	PaDiM [8]	76.6±3.1	88.1±1.7	88.2±1.1	62.8±5.4	68.3±4.0	75.3±1.2
	PatchCore [31]	83.4±3.0	92.2±1.5	90.5±1.5	79.9±2.9	82.8±2.3	81.7±1.6
	WinCLIP+ (ours)	**93.1±2.0**	**96.5±0.9**	**93.7±1.1**	**83.8±4.0**	**85.1±4.0**	**83.1±1.7**
2-shot	SPADE [7]	82.9±2.6	91.7±1.2	91.1±1.0	80.7±5.0	82.3±4.3	81.7±2.5
	PaDiM [8]	78.9±3.1	89.3±1.7	89.2±1.1	67.4±5.1	71.6±3.8	75.7±1.8
	PatchCore [31]	86.3±3.3	93.8±1.7	92.0±1.5	81.6±4.0	84.8±3.2	82.5±1.8
	WinCLIP+ (ours)	**94.4±1.3**	**97.0±0.7**	**94.4±0.8**	**84.6±2.4**	**85.8±2.7**	**83.0±1.4**
4-shot	SPADE [7]	84.8±2.5	92.5±1.2	91.5±0.9	81.7±3.4	83.4±2.7	82.1±2.1
	PaDiM [8]	80.4±2.5	90.5±1.6	90.2±1.2	72.8±2.9	75.6±2.2	78.0±1.2
	PatchCore [31]	88.8±2.6	94.5±1.5	92.6±1.6	85.3±2.1	87.5±2.1	84.3±1.3
	WinCLIP+ (ours)	**95.2±1.3**	**97.3±0.6**	**94.7±0.8**	**87.3±1.8**	**88.8±1.8**	**84.2±1.6**

Experiment

- Few-shot Anomaly Segmentation
  - Few-shot setting

Anomaly Segmentation		MVTec-AD			VisA		
Setup	Method	pAUROC	PRO	F_1-max	pAUROC	PRO	F_1-max
1-shot	SPADE [7]	91.2±0.4	83.9±0.7	42.4±1.0	95.6±0.4	84.1±1.6	35.5±2.2
	PaDiM [8]	89.3±0.9	73.3±2.0	40.2±2.1	89.9±0.8	64.3±2.4	17.4±1.7
	PatchCore [31]	92.0±1.0	79.7±2.0	50.4±2.1	95.4±0.6	80.5±2.5	38.0±1.9
	WinCLIP+ (ours)	**95.2±0.5**	**87.1±1.2**	**55.9±2.7**	**96.4±0.4**	**85.1±2.1**	**41.3±2.3**
2-shot	SPADE [7]	92.0±0.3	85.7±0.7	44.5±1.0	96.2±0.4	85.7±1.1	40.5±3.7
	PaDiM [8]	91.3±0.7	78.2±1.8	43.7±1.5	92.0±0.7	70.1±2.6	21.1±2.4
	PatchCore [31]	93.3±0.6	82.3±1.3	53.0±1.7	96.1±0.5	82.6±2.3	41.0±3.9
	WinCLIP+ (ours)	**96.0±0.3**	**88.4±0.9**	**58.4±1.7**	**96.8±0.3**	**86.2±1.4**	**43.5±3.3**
4-shot	SPADE [7]	92.7±0.3	87.0±0.5	46.2±1.3	96.6±0.3	87.3±0.8	43.6±3.6
	PaDiM [8]	92.6±0.7	81.3±1.9	46.1±1.8	93.2±0.5	72.6±1.9	24.6±1.8
	PatchCore [31]	94.3±0.5	84.3±1.6	55.0±1.9	96.8±0.3	84.9±1.4	43.9±3.1
	WinCLIP+ (ours)	**96.2±0.3**	**89.0±0.8**	**59.5±1.8**	**97.2±0.2**	**87.6±0.9**	**47.0±3.0**

Table 4. Comparison of anomaly segmentation (AS) performance on MVTec-AD and VisA benchmarks. We report the mean and standard deviation over 5 random seeds for each measurement. Bold indicates the best performance.



(a) MVTec-AD (1-shot)

Experiment

- Comparison with many-shot methods

Methods	Setup	AC	AS
WinCLIP (ours)	0-shot	91.8	85.1
WinCLIP+ (ours)	1-shot	93.1	95.2
WinCLIP+ (ours)	4-shot	95.2	96.2
DifferNet [32]	16-shot	87.3	-
TDG [39]	10-shot	78.0	-
RegAD-L [14]	2-shot	81.5	93.3
RegAD [14]	4 + agg.	88.2	95.8
MKD [35]	full-shot	87.7	90.7
P-SVDD [49]	full-shot	92.1	95.7
CutPaste [20]	full-shot	95.2	96.0
PatchCore [31]	full-shot	99.6	98.2

Experiment

- Ablation study – Anomaly Segmentation
  - Patch-token
    - Image encoder (VIT)의 마지막 layer의 patch feature를 그대로 text feature와 비교하여 anomaly score를 계산
  - Image tiling
    - Window를 제외한 부분을 masking하는 대신, window 부분만 따로 추출하여 resizing
    - 준수한 성능을 보이지만, inference time이 매우 긴 것을 확인할 수 있음
  - Harmonic avg.
    - 실험적으로 Harmonic Average를 사용했을 때, 유의미한 성능 향상을 가지는 것을 확인할 수 있음

Method	pAUROC	PRO	F_1-max	Time (ms)
Patch-token	22.4	2.3	8.0	**95.5±18.8**
Image tiling	77.9	57.5	25.5	1442.1±62.2
WinCLIP (ours)	**85.1**	**64.6**	**31.7**	389.4±18.5
w/o image-scale	82.0	63.0	29.5	378.6±20.2
w/o mid-scale	84.0	61.6	30.5	190.7±13.9
w/o small-scale	84.7	63.6	30.6	265.4±15.9
w/o Harmonic avg.	81.5	60.5	27.3	279.9±22.8

Conclusion

- Vision-Language Model인 CLIP을 추가적으로 학습하지 않고 anomaly detection에 적용할 수 있는 방법을 최초로 제안
- Window 방식을 적용 → local dense feature를 획득할 수 있어 Zero-/Few-shot anomaly segmentation을 효과적으로 수행할 수 있음

- Few-normal shot에 대한 feature를 저장하는 방식을 사용하기 때문에, 마찬가지로 new class의 수가 증가하면 memory 이슈가 여전히 발생할 것으로 보임
- Anomaly segmentation task를 수행할 때 평균 inference time이 약 400ms로, 산업에서 쓰기엔 아직 느리다는 한계가 있음

감사합니다